I claim:

1. An approximate message authentication code generated by a cryptography device and comprising a probabilistic checksum generated using as input a message and a shared key and which provides absolute authentication for an origin of the message and approximate integrity for the content of the message.

2. The approximate message authentication code of claim 1, wherein the probabilistic checksum also uses an initial value as input.

3. The approximate message authentication code of claim 1, wherein data in the message are permuted.

4. The approximate message authentication code of claim 3, wherein all of the data in the message are permuted.

5. The approximate message authentication code of claim 3, wherein less than all of the data in the message are permuted.

6. The approximate message authentication code of claim 5, wherein a pseudo-random function is used to permute the data.

7. The approximate message authentication code of claim 5, wherein a random sample of data in the message are permuted.

8. The approximate message authentication code of claim 5, wherein at least one of statistical data and averages of data in the message are permuted.

9. The approximate message authentication code of claim 3, wherein the permuted data are masked into an unbiased, independent, identically distributed set of bits.

10. The approximate message authentication code of claim 3, wherein the permuted data are masked.

11. The approximate message authentication code of claim 10, wherein the data are masked using stream encryption.

12. The approximate message authentication code of claim 10, wherein:

a. the masked data are taken in groups of rows, each row having columns of $|A|$ bits, where A is an integer, and a majority bit value is determined for each column in each group of rows to generate a new group of rows of majority bits; and

b. a majority bit value is determined for each of $|A|$ columns of the new group of rows of majority bits.

13. The approximate message authentication code of claim 12, wherein the groups of rows are all groups of T rows, where T is an integer.

14. The approximate message authentication code of claim 12, wherein all of the groups of rows do not have the same number of rows.

15. The approximate message authentication code of claim 1, wherein the message is a received message and a value of the approximate message authentication code is compared with a value of second approximate message authentication code received with the message.

16. The approximate message authentication code of claim 15, wherein the received message is determined to have acceptable integrity if the approximate message authentication code is one of (1) a same value as and (2) within an acceptable distance of, the second approximate message authentication code.

17. A method performed by a cryptography device for generating an approximate message

authentication code, comprising the steps of:

a. receiving a message containing data and arranging the data into a table having $|A|$ columns and T^2 rows, where A and T are integers;

b. permuting at least some of the arranged data;

c. masking the permuted data;

d. copying the permuted and masked data into T S-arrays, each S-array having $|A|$ columns, and determining a majority bit value of each of the $|A|$ columns for each of the T S-arrays;

e. using the determined majority bits to create a T-array having $|A|$ columns and T rows; and

f. determining the majority value of each of the $|A|$ columns in the T array.

18. The method of claim 17, further comprising the step of generating a pseudo-random bit string before the step of permuting.

19. The method of claim 18, wherein the step of generating the pseudo-random bit string (PRBS) further comprises using a shared key and pseudo-random number generator to generate the PRBS.

20. The method of claim 19, wherein the step of generating the PRBS further comprises using an initial value to generate the PRBS.

21. The method of claim 17, further comprising the step of selecting T to be an odd integer.

22. The method of claim 17, further of comprising the step of selecting a length of $|A|$.

23. The method of claim 17, wherein the step of permuting further comprises permuting the data by row.

24. The method of claim 23, further comprising the step of for each permuted row, permuting

data within each row.

25. The method of claim 24, wherein the step of permuting data within each row further comprises the step of circularly shifting each permuted row a pseudo-random number of places.

26. The method of claim 17, wherein the step of permuting further comprises permuting the data by bit.

27. The method of claim 17, wherein the step of permuting further comprises selecting an unpredictable permutation.

28. The method of claim 27, wherein the step of selecting an unpredictable permutation further comprises using one of a block cipher and a conventional message authentication code.

29. The method of claim 17, wherein the step of permuting comprises permuting all of the data in the message

30. The method of claim 17, wherein the step of permuting comprises permuting less than all of the data in the message.

31. The method of claim 30, wherein the step of permuting further comprises using a pseudo-random function to select the data for permuting.

32. The method of claim 30, wherein the step of permuting further comprises permuting a random sample of data in the message.

33. The method of claim 30, wherein the step of permuting further comprises permuting at least one of statistical data and averages of data in the message.

34. The method of claim 17, wherein the step of masking further comprises the step of stream encrypting the permuted data.

35. The method of claim 17, wherein the step of masking further comprises bitwise exclusive-ORing the permuted data and at least a portion of the pseudo-random bit stream.

36. The method of claim 17, wherein the step of masking comprises generating an unbiased, independent, identically distributed set of 1s and 0s.

37. The method of claim 17, wherein the step of copying the permuted and masked data into S-arrays further comprises selecting each S-array to have T rows.

38. The method of claim 37, wherein the step of copying the permuted and masked data into S-arrays further comprises adding a row of pseudo-random bits to the S-array if T is an even number.

39. The method of claim 17, wherein the step of copying the permuted and masked data into S-arrays further comprises not selecting each S-array to have the same number of rows.

40. A device for generating an approximate message authentication code, comprising:

a. a pseudo-random bit string generator module configured to receive as input a secret key and to output a string of pseudo-random bits;

b. an arrangement module configured to receive a message containing data and arrange the data into a table having $|A|$ columns and T^2 rows, where A and T are integers;

c. a permuting module responsive to the arranged data and at least a portion of the string of pseudo-random bits and configured to permute arranged data;

d. a masking module responsive to the permuting module and at least a portion of the string of pseudo-random bits and configured to mask the randomized data; and

e. a majority module responsive to the masking module and configured to:

i. copy the masked data into T S-arrays, each array having $|A|$ columns, and to determine the majority bit value of each of the $|A|$ columns for each of the S-arrays;

ii. use the determined majority bits to create a T-array having $|A|$ columns and T rows; and

iii. determine the majority value of each of the $|A|$ columns in the T array.

41. The device of claim 40, wherein the pseudo-random bit string generator module further comprises a pseudo-random number generator.

42. The device of claim 41, wherein the pseudo-random number generator is a cryptographically secure random number generator.

43. The device of claim 40, wherein the pseudo-random bit string generator module is further configured to receive as input an initial value.

44. The device of claim 40, wherein T is an odd integer.

45. The device of claim 40, wherein $|A|$ is selected to have a predetermined length, the selected length depending on a sensitivity to bit changes.

46. The device of claim 40, wherein the permuting module is configured to permute the data by row.

47. The device of claim 46, wherein the permuting module is further configured, for each permuted row, to permute data within each row.

48. The device of claim 47, wherein the permuting module is further configured to circularly shift each permuted row a pseudo-random number of places.

49. The device of claim 40, wherein the permuting module is configured to permute the data by one of bit, byte, and data word.

50. The device of claim 40, wherein the permuting module is further configured to use a collision-free, unpredictable permutation to permute the data.

51. The device of claim 40, wherein the permuting module is configured to permute all of the data in the message

52. The device of claim 40, wherein the permuting module is configured to permute less than all of the data in the message.

53. The device of claim 52, wherein permuting module uses a pseudo-random function to select the data for permuting.

54. The device of claim 52, wherein the permuting module is configured to permute a random sample of data in the message.

55. The device of claim 52, wherein the permuting module is configured to permute at least one of statistical data and averages of data in the message.

56. The device of claim 40, wherein the masking module further comprises an exclusive-OR circuit responsive to the permuting module and at least a portion of the string of pseudo-random bits.

57. The device of claim 40, wherein the majority module is configured to copy the masked data into S-arrays each having the same number of rows.

58. The device of claim 40, wherein the majority module is configured to copy the masked data into S-arrays not all having the same number of rows.

59. A method performed by a cryptography device for determining whether a received message from a sender to a recipient has sufficient integrity to accept as an original message sent by the sender:

a. the recipient receiving the received message and a first approximate message authentication code generated by the sender on the original message;

b. the recipient generating a second approximate message authentication code from the received message;

c. comparing the first and second approximate message authentication codes; and

d. determining that the received message has sufficient integrity if one of:

i. the first and second approximate message authentication codes are the same; and

ii. the first and second approximate message authentication codes have no more than a predetermined acceptable number of bit differences.

60. A method performed by a cryptography device for determining an acceptable number of bit differences between a first approximate message authentication code (AMAC) for an original message and a second AMAC for a received message purporting to be the original message, the method comprising the steps of:

a. determining a first expected number of bit differences between the original message and the received message; and

b. converting the first expected number of bit differences into a second expected number of bit differences between the first AMAC and the second AMAC.

61. The method of claim 60, wherein the step of determining further comprises determining an error rate of a transmission or storage medium; and the step of converting further comprises:

a. using the expected error rate, determining a likelihood that bit differences will occur in a column of an instance of an S-array in the second AMAC;

b. determining a distribution of Hamming weights of columns of an instance of an S-array in the second AMAC;

c. using the distribution of Hamming weights, determining a probability that *d* differences in a column of an array will cause a change in a majority bit value in that column;

d. using the likelihood that bit differences will occur in a column of an instance of an S-array in the second AMAC and the probability that *d* differences in a column of an array will cause a change in a majority bit value in that column, determining an expected number of differences between a *T*-array for the first AMAC and a *T*-array for the second AMAC;

e. estimating a distribution of the *d* differences across columns of the *T*-array for the second AMAC; and

f. estimating for a second time the distribution of the *d* differences across columns of the *T*-array for the second AMAC.

62. The method of claim 61, wherein the step of determining a likelihood that bit differences will occur in a column of an instance of an S-array in the second AMAC further comprises using a hyper-geometric distribution.

63. The method of claim 61, wherein the step of determining a distribution of Hamming weights of columns of an instance of an S-array further comprises determining a binomial distribution.

64. The method of claim 61, wherein the step of determining a probability that *d* differences in a column of an array will cause a change in a majority bit value in that column further comprises by determining all cases that can possibly change the majority bit value.

65. The method of claim 61, wherein the step of estimating a distribution of the *d* differences across columns of the *T*-array for the second AMAC further comprises using Bose-Einstein occupancy statistics.